CHAPTER 2

Configuring User Profiles and CSS Parameters

This chapter describes how to configure user profiles and CSS parameters. This chapter also contains information on using the Content API and Command Scheduler features. Information in this chapter applies to all models of the CSS except where noted.

This chapter contains the following sections:

- Configuring User Profiles
- Boot Configuration Mode Commands
- Configuring Host Name
- Configuring Idle Timeout
- Configuring the CSS as a Client of a RADIUS Server
- Controlling Remote Access to the CSS
- Restricting Console, FTP, SNMP, Telnet, XML, and Web Management Access to the CSS
- Configuring Flow Parameters
- Finding an IP Address
- Configuring Content API
- Configuring the Command Scheduler

Configuring User Profiles

The CSS contains a default-profile that resides in the scripts directory on the Internal Disk Module (IDM). This file contains settings that are user-specific; that is, they apply uniquely to each user when the user logs in.

You can customize the following settings for each user:

- CLI prompt
- Expert mode
- History buffer
- Terminal parameters, including idle time, length, more, netmask format, and timeout

Though the settings are user-specific, each default setting applies to all users until the user saves the default-profile to a *username*-profile (where *username* is the current login username). You may choose to continue using the default-profile so that all users logging into a CSS use the same settings. Refer to "Copying and Saving User Profiles" in this chapter for information on saving the default-profile.

If you change a user setting and want to save it in the scripts directory of the current ADI, use a **copy profile** command. If you do not, the CSS stores the setting temporarily in a running-profile. If you attempt to log out of the CSS without saving profile changes, the CSS prompts you that profile changes have been made and allows you to save or discard the changes.

When you upgrade the ADI, user profiles, which are saved in the current ADI directory, are deleted. If you wish to save user profiles permanently, use the **save_profile** command. This command saves the profiles in both the scripts and archive directories in the current ADI. The archive directory is not overwritten during a software upgrade.

To access the CSS IDM, FTP into the CSS. Use the appropriate commands to access the scripts directory and list the contents of the default-profile. When logged into the CSS, use the **show profile** command to display either the default-profile or your *username*-profile.

For example:

```
# show profile

@prompt CSS11150
@no expert
alias all reboot "@configure;boot;rebo"
alias all shutdown "@configure;boot;shutd"
alias all logon "@configure;logging line \${LINE};exit"
alias all logoff "@configure;no logging line \${LINE};exit"
alias all aca-load "@script play service-load"
alias all dnslookup "@script play dnslookup"
alias super save_config "copy running-config startup-config;archive
startup-config"
alias super setup "script play setup"
alias super upgrade "script play upgrade"
alias super monitor "script play monitor"
alias super save_profile "copy profile user-profile;archive script
admin-profile
"
set CHECK_STARTUP_ERRORS "1" session
```

This section contains information on:

- Configuring User Terminal Parameters
- Using Expert Mode
- Changing the CLI Prompt
- Modifying the History Buffer
- Copying and Saving User Profiles

Configuring User Terminal Parameters

To configure terminal parameters, use the **terminal** command. These parameters control output to the system terminal screen. Terminal parameters are user-specific; that is, they apply uniquely to each CSS user.

Use the **copy profile user-profile** command to add terminal command parameters to your user profile so that the parameters are used each time you log in. Otherwise you must reenter the commands for the parameters to take effect each time you log in.

The options for this command are:

- **terminal idle** - Set the session idle timer.
- **terminal length** - Set the terminal screen output length.
- **terminal more** - Enable terminal more support. The default is enabled.
- **terminal netmask-format** - Control subnet mask display.
- **terminal timeout** - Set the session maximum login time.

Configuring Terminal Idle

To set the time a session can be idle before the CSS terminates a console or Telnet session, use the **terminal idle** command. The default value is 0 (disabled). This command is available at the User and SuperUser prompts. Enter an idle time between 0 and 65535 minutes.

To set a terminal idle time, enter:

```
# terminal idle 15
```

To revert the terminal idle time to its default of disabled, enter:

```
# no terminal idle
```

Configuring Terminal Length

To set the number of output lines the CLI displays on the terminal screen, use the **terminal length** command. This command is available at the User and SuperUser prompts. Enter the number of lines you want the CLI to display from 2 to 65535. The default is 25 lines.

To set the line number to 35, enter:

```
# terminal length 35
```

To set the number of lines to the default of 25 lines, enter:

```
# no terminal length
```

Configuring Terminal More

To enable support for **more** terminal functions, use the **terminal more** command. This command is available at the User and SuperUser prompts. You can also toggle the **more** function on and off within a session by using the ESC-M key sequence.

To enable **more** terminal functions, enter:

```
# terminal more
```

To disable support for **more** terminal functions, enter:

```
# no terminal more
```

Configuring Terminal Netmask-Format

To determine how the CSS displays subnet masks in show screens, use the **terminal netmask-format** command. This command is available at the User and SuperUser prompts. The options for this command are:

- **terminal netmask-format bitcount** - Displays masks in bitcount (for example, /24).
- **terminal netmask-format decimal** - Displays masks in dotted-decimal format (for example, 255.255.255.0). This is the default format.
- **terminal netmask-format hexadecimal** - Displays masks in hexadecimal format (for example, OXFFFFFFOO).

To display subnet masks in bitcount format, enter:

```
# terminal netmask-format bitcount
```

To revert to the default display format (**decimal**), enter:

```
# no terminal netmask format
```

Configuring Terminal Timeout

To set the total amount of time a session can be logged in before the CSS terminates a console or Telnet session, use the **terminal timeout** command. The default value is 0 (disabled). This command is available at the User and SuperUser prompts. Enter a timeout value between 0 and 65535 minutes.

To set a terminal timeout value, enter:

```
# terminal timeout 30
```

To revert the terminal timeout value to its default (disabled), enter:

```
# no terminal timeout
```

Using Expert Mode

Expert mode allows you to turn the CSS confirmation capability on or off. Expert mode is available at the SuperUser prompt and is **off** by default. When expert mode is off, the CSS prompts you for confirmation when you:

- Execute commands that could delete or change operating parameters
- Exit a terminal session (console or Telnet) without copying the running-config to startup-config
- Create services, owners, and content rules

Turning expert mode on *prevents* the CSS from prompting you for confirmation when you make configuration changes. To prevent the CSS from prompting you for confirmation when you make configuration changes, enter:

```
# expert
```

To allow the CSS to prompt you for confirmation when you make configuration changes, enter:

```
# no expert
```

For example, when you issue the command to create an owner and expert mode is off, the CSS prompts you to verify the command, enter:

```
(config)# owner arrowpoint.com
Create owner <arrowpoint.com>, [y/n]:y
(config-owner[arrowpoint.com])#
```

Changing the CLI Prompt

The CLI default prompt displays as the product model number followed by the # symbol. The CSS adds a # sign to the prompt automatically to indicate SuperUser mode. To change the default prompt, enter the **prompt** command as shown in the following example (maximum of 15 alphanumeric characters):

```
CSS11800# prompt CSS1-lab
CSS1-lab#
```

To save the new prompt, add it to user or default profiles. To restore the prompt to its default, use the **no prompt** command.

Modifying the History Buffer

Use the **history** command to modify the history buffer length. The command line history buffer stores the most recent CLI commands that you enter. Enter the number of lines you want in the history buffer as an integer from 0 to 256. The default is 20. This command is available in SuperUser mode.

To set the history buffer to 80 lines, enter:

```
# history length 80
```

To disable the history function (setting of 0), enter:

```
# history length 0
```

To restore the history buffer to the default of 20 lines, enter:

```
# no history length
```

Displaying the History Buffer

Use the **show history** command to display the history buffer. The history buffer is cleared automatically upon reboot.

For example:

```
# show history

    history
    show history
    show ip routes
    show ip summary
    show ip stat
    clock
    clock date
    clock time
    show history
```

Copying and Saving User Profiles

Use the **copy profile** command to copy the running profile from the CSS to the default-profile, an FTP server, a TFTP server, or your user-profile. The options are:

- **copy profile default-profile** - Copy the running profile to the default profile
- **copy profile user-profile** - Copy the running profile to your user profile
- **copy profile ftp** - Copy the running profile to an FTP server
- **copy profile tftp** - Copy the running profile to a TFTP server

Note If you exit the CSS without copying changes in the running profile to your *username*-profile or default-profile, the CSS prompts you that the profile has changed and queries whether or not you want to save your changes. If you respond with **y**, the CSS copies the running profile to your *username*-profile or the default-profile.

Refer to the following sections for information on these options.

Copying the Running Profile to the Default-Profile

Use the **copy profile default-profile** command to copy the running profile to the default profile. This command is available at the SuperUser prompt.

For example, enter:

```
# copy profile default-profile
```

Copying the Running Profile to a User Profile

Use the **copy profile user-profile** command to proactively copy the changes made to the running profile to the user profile. This command creates a file *username*-profile if one does not exist (where *username* is the current username).

For example, enter:

```
# copy profile user-profile
```

Copying the Running Profile to an FTP Server

Use the **copy profile ftp** command to copy the running profile to an FTP server. The syntax is:

copy profile ftp *ftp_record filename*

The variables are:

- *ftp_record* - The name of the FTP record file that contains the server IP address, username, and password. Enter an unquoted text string with no spaces and a maximum length of 32 characters.
- *filename* - The name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces.

For example, enter:

```
# copy profile ftp arrowrecord \records\arrowftprecord
```

Copying the Running Profile to a TFTP Server

Use the **copy profile tftp** command to copy the running profile to a TFTP server. The syntax is:

copy profile tftp *ip_or_host filename*

The variables are:

- *ip_address* or *host* - The IP address or host name of the server to receive the file. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *filename* - The name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum length of 32 characters.

For example, enter:

```
# copy profile tftp 192.168.3.6 \home\bobo\bobo-profile
```

Boot Configuration Mode Commands

Boot configuration mode contains all of the commands necessary to manage booting the CSS and to maintain the software revision. To access this mode, use the **boot** command from global configuration mode. The prompt changes to (config-boot).

To access boot mode, enter:

```
(config)# boot
```

The CSS enters into boot mode.

```
(config-boot)#
```

For information about commands available in boot mode, refer to the following sections:

- Unpacking an ArrowPoint Distribution Image (ADI)
- Removing an ArrowPoint Distribution Image (ADI)
- Specifying the Primary BOOT Configuration
- Specifying the Secondary Boot Configuration
- Configuring a Boot Configuration Record for the Passive SCM
- Showing the BOOT Configuration
- Booting the CSS from a Network Drive

Unpacking an ArrowPoint Distribution Image (ADI)

Use the **unpack** command to unpack the ArrowPoint Distribution Image (ADI) on the CSS disk. Enter the ADI filename as an unquoted text string with a maximum length of 32 characters. For example, enter:

```
(config-boot)# unpack ap0500002.adi
```

Note Before unpacking the ADI, you must first copy the ADI to the CSS disk. Use the **copy ftp ftp_record** *filename* **boot-image** command to copy the ADI to the CSS disk.

Removing an ArrowPoint Distribution Image (ADI)

Use the **remove** command to remove an ArrowPoint Distribution Image (ADI) that is not currently running on the CSS. To display a list of ADIs installed on your CSS, enter **remove ?**. To display the ADI you are currently running, use the **version** command.

Enter the ADI filename as an unquoted text string with a maximum length of 32 characters.

For example, to remove an ADI, enter:

```
(config-boot)# remove ap0410008
```

Specifying the Primary BOOT Configuration

Use the **primary** command to specify the primary boot configuration. The options for this boot mode command are:

- **primary boot-file** - Specify the primary boot file
- **primary boot-type** - Specify the primary boot method, local disk, using FTP, or a network-mounted file system using FTP
- **primary config-path** - Specify the path to a network CSS configuration

Refer to the following sections for more information on these options and associated variables.

Configuring the Primary Boot-File

Use the **primary boot-file** command to specify the primary boot file. Enter the primary boot file as an unquoted text string with no spaces and a maximum length of 64 characters.

To specify the primary boot filename, enter:

```
(config-boot)# primary boot-file ap0500002
```

To display a list of boot filenames, enter:

```
(config-boot)# primary boot-file ?
```

To remove the primary boot file, enter:

```
(config-boot)# no primary boot-file
```

Configuring the Primary Boot-Type

Use the **primary boot-type** command to specify the primary boot method, either from the local disk or using FTP. The syntax and options for this boot mode command are:

- **primary boot-type boot-via-disk** - Boot the CSS from software currently on the IDM.
- **primary boot-type boot-via-ftp** *ftp_record* - Download an ADI file containing CSS software that you want to install on the IDM. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies it to the IDM, and unpacks it.
- **primary boot-type boot-via-network** *ftp_record* - Use FTP to boot the CSS from software located on a network-mounted file system on a remote system (such as a PC or UNIX workstation). The CSS boots independently from the IDM and loads the configuration into memory. Instead of the CSS disk, the network file system contains the CSS software.

Enter the *ftp_record* as the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces.

For example, to configure the primary boot-type to **boot-via-disk**, enter:

```
(config-boot)# primary boot-type boot-via-disk
```

To remove the primary boot type, enter:

```
(config-boot)# no primary boot-type
```

Configuring the Primary Config-Path

Use the **primary config-path** command to specify the alternate path to a network configuration for the network boot method. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server (such as a PC or UNIX workstation) as defined in the FTP record for the network boot method.

When using an alternate configuration path, make sure that the path leads to a directory containing the script, log, and info subdirectories and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories, then copy the files from the boot image to the subdirectories.

Enter the configuration pathname as an unquoted text string with no spaces and a maximum length of 64 characters.

To configure the primary config path, enter:

```
(config-boot)# primary config-path f:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no primary config-path
```

Specifying the Secondary Boot Configuration

Use the **secondary** command to specify the secondary boot configuration. The secondary boot configuration is used when the primary configuration fails. The options for this boot mode command are:

- **secondary boot-file** - Specify the secondary boot file
- **secondary boot-type** - Specify the boot method, local disk or FTP
- **secondary config-path** - Specify the path to a network configuration using FTP

For more information on these options and associated variables, refer to the following sections.

Specifying the Secondary Boot-File

Use the **secondary boot-file** command to specify the secondary boot file that the CSS uses when the primary boot configuration fails. Enter the boot file as an unquoted text string with no spaces and a maximum length of 64 characters.

To specify the secondary boot filename, enter:

```
(config-boot)# secondary boot-file ap0410008
```

To display a list of secondary boot filenames, enter:

```
(config-boot)# secondary boot-file ?
```

To remove the secondary boot file, enter:

```
(config-boot)# no secondary boot-file
```

Specifying the Secondary Boot-Type

Use the **secondary boot-type** command to boot the system using the local disk, FTP, or a network-mounted file system. The FTP record contains the IP address, username, and password for the FTP server. Enter the *ftp_record* as an unquoted text string with no spaces.

The syntax and options for this boot mode command are:

- **secondary boot-type boot-via-disk** - Boot the system from local disk.
- **secondary boot-type boot-via-ftp** *ftp_record* - Download an ADI file containing CSS software that you want to install on the IDM. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies it to the IDM, and unpacks it.
- **secondary boot-type boot-via-network** *ftp_record* - Use FTP to boot the CSS from software located on a network-mounted file system on a remote system (such as a PC or UNIX workstation). The CSS boots independently from the IDM and loads the configuration into memory. Instead of the CSS disk, the network file system contains the CSS software.

For example, to specify the secondary boot type as **boot-via-disk**, enter:

```
(config-boot)# secondary boot-type boot-via-disk
```

To remove the secondary boot type, enter:

```
(config-boot)# no secondary boot-type
```

Specifying the Secondary Config-Path

Use the **secondary config-path** command to specify the alternate path to a network configuration for the network boot method. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server (such as a PC or UNIX workstation) as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure that the path leads to a directory containing the script, log, and info subdirectories and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories, then copy the files from the boot image to the subdirectories.

Enter the configuration pathname as an unquoted text string with no spaces and a maximum length of 64 characters.

To configure the secondary config path, enter:

```
(config-boot)# secondary config-path f:/bootdir/
```

To remove the secondary network configuration path, enter:

```
(config-boot)# no secondary config-path
```

Configuring a Boot Configuration Record for the Passive SCM

Use the **passive** command to configure the boot configuration record for the current passive SCM installed in a CSS 11800. The boot configuration record consists of the IP address, subnet mask, boot method, and boot file.

With the **sync** option for this command, you can copy the boot configuration record from the active SCM to the passive SCM. In most CSS configurations, the active and passive SCMs will have the same boot record.

This command also allows you to configure the individual components of the boot configuration record on the passive SCM. For example, you can configure a boot record on the passive SCM that has a software version that differs from the active SCM. This allows you run a new software version on the active SCM with the security of having an older software version on the passive SCM.

You can also configure a different IP address on the passive SCM to track an active-to-passive state transition between the SCMs. You can accomplish this through a network management station where you can receive SNMP host traps.

Note The **passive** command and its options only affect the current passive SCM. When you configure the passive SCM, the set values are loaded into its nonvolatile RAM. If the passive SCM transitions to the active state, it continues to retain these values but is no longer affected by these commands; boot commands are not saved in the running-config.

The options for this boot mode command are:

- **passive ip address** - Configure the system boot IP address for the passive SCM.
- **passive primary boot-file** - Specify the primary boot file for the passive SCM.
- **passive primary boot-type** - Specify the primary boot method, local disk, FTP, or network-mounted file system using FTP, for the passive SCM.
- **passive primary config-path** - Specify the primary alternate path to a network CSS configuration for the passive SCM.
- **passive secondary boot-file** - Specify the secondary boot file for the passive SCM.
- **passive secondary boot-type** - Specify the secondary boot method, local disk, FTP, or network-mounted file system via FTP, for the passive SCM.
- **passive secondary config-path** - Specify the secondary alternate path to a network CSS configuration for the passive SCM.
- **passive subnet mask** - Configure the system boot subnet mask for the passive SCM.
- **passive sync** - Copy the boot configuration record from the active SCM to the passive SCM.

For more information on these options and associated variables, refer to the following sections.

Configuring the Passive SCM IP Address

Use the **passive ip address** command to configure the system boot IP address for the passive SCM. Enter the IP address for the passive SCM that will be used on boot up. Do not enter an all zero IP address.

For example, enter:

```
(config-boot)# passive ip address 172.16.3.6
```

To change the passive SCM boot IP address, reissue the **passive ip address** command.

Configuring the Passive SCM Primary Boot File

Use the **passive primary boot-file** command to specify the primary boot image for the passive SCM. Enter the filename of the primary boot image for the passive SCM as an unquoted text string with no spaces and a maximum length of 64 characters. To display a list of filenames, enter **passive primary boot-file ?**.

For example, enter:

```
(config-boot)# passive primary boot-file ap0500002
```

To remove the primary boot file from the passive SCM, enter:

```
(config-boot)# no passive primary boot-file
```

Configuring the Passive SCM Primary Boot Type

Use the **passive primary boot-type** command to specify the primary boot method, the local disk, FTP, or a network-mounted file system for the passive SCM. The syntax and options for this boot mode command are:

- **passive primary boot-type boot-via-disk** - Boot the system from local disk.
- **passive primary boot-type boot-via-ftp** *ftp_record* - Download an ADI file containing CSS software that you want to install on the IDM. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies it to the passive SCM, and unpacks it.
- **passive primary boot-type boot-via-network** *ftp_record* - Use FTP to boot the CSS from software located on a network-mounted file system on a remote system (such as a PC or UNIX workstation). The CSS boots independently from the passive SCM and loads the configuration into memory. Instead of the CSS disk, the network file system contains the CSS software.

Enter the *ftp_record* as the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces.

For example, enter:

```
(config-boot)# passive primary boot-type boot-via-ftp arecord
```

To remove the primary boot type from the passive SCM, enter:

```
(config-boot)# no passive primary boot-type
```

Configuring the Passive SCM Primary Configuration Path

Use the **passive primary config-path** command to specify the alternate path to a network configuration for the network boot method for the passive SCM. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server (such as a PC or UNIX workstation) as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure that the path leads to a directory containing the script, log and info subdirectories, and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories in the unZipped boot image. First, create these subdirectories. Then copy the files from the boot image to the subdirectories.

Enter the configuration path for network configuration. Enter an unquoted text string with no spaces and a maximum length of 64 characters. For example, enter:

```
(config-boot)# passive primary config-path c:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no passive primary config-path
```

Configuring the Passive SCM Secondary Boot File

Use the **passive secondary boot-file** command to specify the secondary boot image for the passive SCM. Enter the boot file name for the primary boot image as an unquoted text string with no spaces and a maximum length of 64 characters. To display a list of boot filenames, enter **passive secondary boot-file ?**. For example:

```
(config-boot)# passive secondary boot-file ap0410008
```

To remove the secondary boot file from the passive SCM, enter:

```
(config-boot)# no passive secondary boot-file
```

Configuring the Passive SCM Secondary Boot Type

Use the **passive secondary boot-type** command to boot the system using the local disk, FTP, or a network-mounted file system for the passive SCM. The syntax and options for this boot mode command are:

- **passive secondary boot-type boot-via-disk** - Boot the system from local disk.
- **passive secondary boot-type boot-via-ftp** *ftp_record* - Download an ADI file containing CSS software that you want to install on the passive SCM. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, and unpacks it.
- **passive secondary boot-type boot-via-network** *ftp_record* - Use FTP to boot the CSS from software located on a network-mounted file system on a remote system (such as a PC or UNIX workstation). The CSS boots independently from the passive SCM and loads the configuration into memory. Instead of the CSS disk, the network file system contains the CSS software.

Enter the *ftp_record* as the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces.

For example, enter:

```
(config-boot)# passive secondary boot-type boot-via-disk
```

To remove the secondary boot type from the passive SCM, enter:

```
(config-boot)# no passive secondary boot-type
```

Configuring the Passive SCM Secondary Configuration Path

Use the **passive secondary config-path** command to specify the secondary alternate path to a network configuration for the network boot method for the passive SCM. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server (such as a PC or UNIX workstation) as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure that the path leads to a directory containing the script, log and info subdirectories and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories. Then copy the files from the boot image to the subdirectories.

Enter the configuration path as an unquoted text string with no spaces and a maximum length of 64 characters.

For example, enter:

```
(config-boot)# passive secondary config-path c:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no passive secondary config-path
```

Configuring the Passive SCM Subnet Mask

Use the **passive subnet mask** command to configure the system boot subnet mask for the passive SCM.

For example, enter:

```
(config-boot)# passive subnet mask 255.255.0.0
```

Copying the Boot Configuration Record from the Active SCM to the Passive SCM

Use the **passive sync** command to copy the primary and secondary boot configuration record from the nonvolatile RAM (NVRAM) of the active SCM to its passive SCM backup. This command is available in boot mode.

For example, enter:

```
(config-boot)# passive sync
```

Showing the BOOT Configuration

Use the **show boot-config** command to display your boot configuration. For example:

```
(config-boot)# show boot-config

!********************** BOOT CONFIG **********************
primary boot-file ap0500002
primary boot-type boot-via-disk
subnet mask 255.0.0.0
ip address 172.16.36.58
```

Booting the CSS from a Network Drive

The network booting feature enables you to boot the CSS from a network drive using the .zip file included on your Documentation and System Software compact disc. When you configure the CSS for network boot, the Internal Disk Module (IDM) is not required. To avoid affecting network bandwidth consumption, do not configure logging to disk when booting the CSS from a network drive.

Note Network boot does not support core dumps.

Perform a network boot if:

- You want multiple CSSs to use the same boot image while keeping their own configuration information. Provide an alternate path for the location of the configuration information. This information must exist on the same network file system as the boot image.

  Note When using an alternate configuration path, make sure that the path leads to a directory containing the script, log and info subdirectories. These subdirectories must contain the files in the corresponding subdirectories in the boot image. Create these subdirectories, then copy the files from the boot image.

- The CSS has a hard drive failure. A network boot allows the CSS to boot independently from its hard drive and to load the configuration into memory.

You can configure network boot for CSS 11800:

- Primary SCMs
- Passive SCMs

Configuring Network Boot for a Primary SCM

To configure network boot for a primary SCM:

1. Ensure the SCM management port has access to the network drive from which you are booting the CSS. The SCM will mount the drive, and read and write to it.
2. FTP the software .zip file to the network drive base directory specified in the FTP record. This must be the same directory from which you are booting the CSS.
3. Unzip the file. You must use the .zip distribution format for network loading.
4. Configure the FTP record (refer to the section entitled "Configuring an FTP Record" in Chapter 1, Logging in and Getting Started). Note that the config-path and the base directory path in the ftp-record associated with the network boot must not contain a pathname that collides with a non-network driver name (for example, c: or host:). For example, enter:

   ```
   # ftp-record bootrecord 192.168.19.21 bobo encrypted-password
   "secret" e:/adi_directory/
   ```

   This directory must contain the unzipped files.
5. Configure the CSS to boot from a network drive. For example, enter:

   ```
   (config-boot)# primary boot-type boot-via-network bootrecord
   ```

6. Optionally, configure a primary configuration path to allow multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through the FTP server as defined in the FTP record. For example, enter:

   ```
   (config-boot)# primary config-path e:/adi_directory/
   ```

Configuring Network Boot for a Passive SCM

To configure network boot for a CSS 11800 passive SCM:

1. Configure an FTP record for the passive SCM, if not already configured. Refer to "Configuring a Boot Configuration Record for the Passive SCM" in this chapter.
2. Ensure the passive SCM management port has access to the network drive from which you are booting the CSS. If the primary SCM fails, the passive SCM will connect to the remote disk and load the software configuration.
3. Configure the CSS to boot from a network drive. For example, enter:

   ```
   (config-boot)# passive primary boot-type boot-via-network
   bootrecord
   ```

   To display a list of configured ftp records, reenter the command and use a "?". For example, enter:

   ```
   (config-boot)# passive primary boot-type boot-via-network
   bootrecord ?
   ```

4. Optionally, configure a primary configuration path to allow multiple CSSs to use the same boot image while keeping their configuration information in separate directories. Your FTP daemon must support the drive mapping. Also, the CSS must be able to access the configuration path through the FTP server as defined in the FTP record. For example, enter:

   ```
   (config-boot)# primary config-path e:/adi_directory/
   ```

Showing Network Boot Configurations

To display the network boot configuration, use the **version** command. For example:

```
(config)# version

Version:            ap0500002 (5.00 Build 02)
Network Path:       e:/adi_directory/
Config Path:        e:/adi_directory/
Flash (Locked):     4.10 Build 8
Flash (Operational):4.01 Build 3
Type:               PRIMARY
License Cmd Set:    Standard Feature Set
                    Enhanced Feature Set
                    SSH Server
```

You can also use the **show boot-config** command to display network boot configuration information. For example:

```
(config)# show boot-config

!********************** BOOT CONFIG **********************
secondary config-path  e:/adi_directory/
secondary boot-type    boot-via-network Secondary-Boot
primary boot-file      ap0500002
primary boot-type      boot-via-network
subnet mask            255.0.0.0
ip address             192.168.4.226
```

Configuring Host Name

Use the **host** command to manage entries in the Host table. The Host table is the static mapping of mnemonic host names to IP address, analogous to the ARP table. The syntax for this global configuration mode command is:

host *host_name ip_address*

- *host_name* - The name of the host. Enter an unquoted text string with no spaces and a length of 1 to 16 characters.
- *ip_address* - The address associated with the host name. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

For example, enter:

```
(config)# host CSS11150-LML 192.168.3.6
```

Note To add a host to the Host table, the host name must not already exist. To change a current host address, remove it and then add it again.

To remove an existing host from the Host table, enter:

```
(config)# no host CSS11150-LML
```

To display a list of host names, enter:

```
(config)# show running-config global
```

Configuring Idle Timeout

To globally set the total amount of time all sessions can be active before the CSS terminates a console or Telnet session, use the **idle timeout** command. Enter a timeout value between 0 and 65535 minutes. The default value is enabled for 5 minutes.

Note To override the idle timeout value for a specific session, configure the **terminal timeout command**. Terminal commands are user-specific; that is, they apply uniquely for each CSS user.

It is recommended that you configure the idle timeout to at least 30 minutes. Setting this value to 30 minutes:

- Cleans up idle Telnet sessions
- Helps prevent busy conditions due to a high number of active Telnet sessions

To set an idle timeout value, enter:

```
(config)# idle timeout 15
```

To revert the terminal timeout value to its default of enabled for 5 minutes, enter:

```
(config)# no idle timeout
```

Configuring the CSS as a Client of a RADIUS Server

The Remote Authentication Dial-In User Server (RADIUS) protocol is a distributed client/server protocol that protects networks against unauthorized access. It uses the User Datagram Protocol (UDP) to exchange authentication and configuration information between the CSS authentication client and the active authentication server that contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software.

Use the **radius-server** command to configure the CSS as a client of a RADIUS server for authentication requests by remote or local users who require authorization to access network resources.

When a user remotely logs into a CSS operating as a RADIUS client, the CSS sends an authentication request (including user name, encrypted password, client IP address, and port ID) to the central RADIUS server. The RADIUS server is responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver services to the users. Transactions between the RADIUS client and the RADIUS server are authenticated through the use of a shared secret.

Once the RADIUS server receives the authentication request, it validates the sending client and consults a database of users to match the login request. After the RADIUS server performs user authentication, it transmits one of the following authentication responses back to the RADIUS client:

- **Accept** - The user is authenticated (all conditions are met).
- **Reject** - The user is not authenticated and is prompted to reenter the username and password, or access is denied (the username does not exist in the server's database).

If no response is returned by the RADIUS server within a period of time, the authentication request is retransmitted a predefined number of times (both options are specified in the **radius-server** command). The RADIUS client can forward requests to an alternate secondary RADIUS server in the event that the primary server is down or is unreachable.

In a configuration where both a primary RADIUS server and a secondary RADIUS server are specified, and one or both of the RADIUS servers become unreachable, the CSS automatically transmits a keepalive authentication request to query the server(s). The CSS transmits the username “query” and the password “areyouup” to the RADIUS server (encrypted with the RADIUS server’s key) to determine its state. The CSS continues to send this keepalive authentication request until the RADIUS server indicates that it is available.

Configuring the CSS as a RADIUS Client

Note This section assumes that you have properly configured your RADIUS server implementation. Cisco Systems does not provide RADIUS server software, and it is beyond the scope of this document to cover the different RADIUS server configurations.

Use the **radius-server** command and its options to specify the RADIUS server host (primary RADIUS server, and, optionally, a secondary RADIUS Server), communication time interval settings, and a shared secret text string. This command is available in configuration mode. The options for this command are:

- **radius-server primary** *ip_address* **secret** *string* {**auth-port** *port_number*}- Specify the primary RADIUS server.
- **radius-server secondary** *ip_address* **secret** *string* {**auth-port** *port_number*} - Specify the secondary RADIUS server. Configuration of a secondary RADIUS server is optional.
- **radius-server dead-time** *seconds* - Set the time interval (in seconds) that the CSS probes an inactive RADIUS server (primary and secondary) to determine if it is back online.

- **radius-server retransmit** *number* - Set the number of retransmissions for an authentication request to the RADIUS server.
- **radius-server timeout** *seconds* - Set the time interval the CSS waits before retransmitting an authentication request.

Note After configuring the RADIUS server, enable RADIUS authentication for console and virtual logins (if the user and password pair is not in the local user database) through the **virtual authentication** and **console authentication** commands. Refer to "Controlling Remote Access to the CSS" later in this chapter for details.

Specifying a Primary RADIUS Server

Use the **radius-server primary** command to specify a primary RADIUS server to authenticate user information from the CSS RADIUS client (console or virtual authentication). The syntax for this global configuration mode command is:

radius-server primary *ip_address* **secret** *string* {**auth-port** *port_number*}

Options and variables include:

- **primary** *ip_address* - The IP address or host name for the primary RADIUS server. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).
- **secret** *string* - The shared secret text string between the primary RADIUS server and the CSS RADIUS client. The shared secret allows authentication transactions between the client and primary RADIUS server to occur. Enter the shared secret as a case-sensitive string with no spaces (16 characters maximum).
- **auth-port** *port_number* - Optional. The UDP port on the primary RADIUS server allocated to receive authentication packets from the RADIUS client. Valid entries are 0 to 65535. The default is 1645.

To specify a primary RADIUS server, enter:

```
(config)# radius-server primary 172.27.56.76 secret Hello
auth-port 30658
```

To remove a primary RADIUS server, enter:

```
(config)# no radius-server primary
```

Specifying a Secondary RADIUS Server

Use the **radius-server secondary** command to specify a secondary RADIUS server to authenticate user information from the CSS RADIUS client (console or virtual authentication). The CSS directs authentication requests to the secondary RADIUS server when the specified RADIUS primary server is unavailable. The syntax for this global configuration mode command is:

radius-server secondary *ip_address* **secret** *string* {**auth-port** *port_number*}

Note Configuration of a secondary RADIUS server is optional.

Options and variables include:

- **secondary** *ip_address* - The IP address or host name for the secondary RADIUS server. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).
- **secret** *string* - The shared secret text string between the secondary RADIUS server and the CSS RADIUS client. The shared secret allows authentication transactions between the client and secondary RADIUS server to occur. Enter the shared secret as a case-sensitive string with no spaces (16 characters maximum).
- **auth-port** *port_number* - Optional. The UDP port on the primary RADIUS server allocated to receive authentication packets from the RADIUS client. Valid entries are 0 to 65535. The default is 1645.

To specify a secondary RADIUS server, enter:

```
(config)# radius-server secondary 172.27.56.79 secret Hello
auth-port 30658
```

To remove a secondary RADIUS server, enter:

```
(config)# no radius-server secondary
```

Configuring the RADIUS Server Timeouts

Use the **radius-server timeout** command to specify the time interval that the CSS is to wait for the RADIUS server (primary or secondary) to reply to an authentication request before retransmitting requests to the RADIUS server. You configure the number of retransmitted requests to the server through the **radius-server retransmit** command. Valid entries are 1 to 255 seconds. The default is 10 seconds.

To configure the configure the RADIUS server timeout interval to 1 minute (60 seconds), enter:

```
(config)# radius-server timeout 60
```

To set the RADIUS server retransmit request back to the default of 10 seconds, enter:

```
(config)# no radius-server timeout
```

Configuring the RADIUS Server Retransmits

Use the **radius-server retransmit** command to specify the number of times the CSS is to retransmit an authentication request to a timed-out RADIUS server before considering the server dead and stop transmitting. If a secondary RADIUS server has been identified, that server is selected as the active server. Valid entries are 1 to 30 retries. The default is 3.

If the RADIUS server does not respond to the CSS retransmitted requests, the CSS considers the server as dead, stops transmitting to the server, and starts the dead timer as defined through the **radius-server dead-time** command. If a secondary server is configured, the CSS transmits the requests to the secondary server. If the secondary server does not respond to the request, the CSS considers it dead and starts the dead timer. If there is no active server, the CSS stops transmitting requests until the primary RADIUS server becomes alive.

To configure the number of RADIUS server retransmits to 5, enter:

```
(config)# radius-server retransmit 5
```

To set the RADIUS server retransmit request back to the default of 3 retries, enter:

```
(config)# no radius-server retransmit
```

Configuring the RADIUS Server Dead-Time

Use the **radius-server dead-time** command to set the time interval in which the CSS verifies whether a non-functional server is operational. During the set time interval, the CSS sends probe access-request packets to verify that the RADIUS server (primary or secondary) is available and can receive authentication requests. The dead-time interval starts when the server does not respond to the number of authentication request transmissions configured through the **radius-server retransmit** command. When the server responds to a probe access-request packet, the CSS transmits the authentication request to the server. Valid entries are 1 to 255 seconds. The default is 5 seconds.

To configure the RADIUS server dead-time to 15 seconds, with probe access-requests enabled, enter:

```
(config)# radius-server dead-time 15
```

To set the RADIUS server dead-time request back to the default of 5 seconds, enter:

```
(config)# no radius-server dead-time
```

Showing RADIUS Server Configuration Information

Use the **show radius** command to display information and statistics about the RADIUS server configuration. The syntax and options are:

- **show radius config** [**primary**|**secondary**|**all**] - Display RADIUS configuration information for a specific server or all servers, identified by type.
- **show radius stat** [**primary**|**secondary**|**all**] - Display RADIUS authentication statistics for a specific server or all servers, identified by type.

To view the configuration for a RADIUS primary server, enter:

```
(config)# show radius config primary
```

To view the authentication statistics for a RADIUS secondary server, enter:

```
(config)# show radius stats secondary
```

Table 2-1 describes the fields in the **show radius config** output.

Table 2-1 Field Descriptions for the show radius config Command

Field	Description
Server IP Address	The IP address or host name for the specified RADIUS server.
Secret	The shared secret text string between the specified RADIUS server and the CSS RADIUS client.
Port	The UDP port on the specified RADIUS server allocated to receive authentication packets from the CSS RADIUS client. The default port number is 1645.
State	The operational stats of the RADIUS server (ALIVE, DOWN, UNKNOWN).
Dead Timer	The time interval (in seconds) that the CSS probes a RADIUS server (primary or secondary), which is not responding, to determine if it is operational and can receive authentication requests.
Timeout	The interval (in seconds) the CSS RADIUS client waits for the RADIUS server to reply to an authentication request before retransmitting requests to the RADIUS server.
Retransmit Limit	The number of times the CSS RADIUS client retransmits an authentication request a timed out RADIUS server before stopping transmission to that server.
Probes	The packets that the CSS RADIUS client automatically transmits to determine if the RADIUS server is still available and can receive authentication requests.

Table 2-2 describes the fields in the **show radius stat** output.

Table 2-2 Field Descriptions for the show radius stat Command

Field	Description
Server IP address	The IP address or host name of the specified RADIUS server
Accepts	The number of times the RADIUS server accepts an authentication request from the CSS RADIUS client
Requests	The number of times the CSS RADIUS client issues an authentication request to the RADIUS server
Retransmits	The number of times the CSS RADIUS client retransmits an authentication request to the active RADIUS server after a timeout occurred
Rejects	The number of times the CSS RADIUS client receives a reject notification from the RADIUS server while trying to establish an authentication request
Bad Responses	The number of times the CSS RADIUS client receives a bad transmission from the RADIUS server
Bad Authenticators	The number of times the RADIUS server denies an authentication request from the CSS RADIUS client
Pending Requests	The number of pending authentication requests to the RADIUS server
Timeouts	The number of times the CSS RADIUS client reached the specified timeout interval while waiting for the RADIUS server to reply to an authentication request
Discarded Authentication Requests	The number of authentication requests that were discarded while the primary or secondary RADIUS server was down

Controlling Remote Access to the CSS

To control remote access to the CSS, use the **virtual** command or the **console** command. By using **virtual** commands, you allow users to log into the CSS remotely with or without requiring a username and password, or you can deny all remote access to users. Telnet, FTP, SSHD, and the Device Management user interface are examples of remote access. By using **console** commands, you specify whether console port authentication of locally-defined usernames and passwords logging into the CSS is enabled.

Note Before you can use RADIUS as either the virtual authentication method or the console authentication method, you must enable communication with the RADIUS security server using the **radius-server** command (refer to "Configuring the CSS as a Client of a RADIUS Server" earlier in this chapter for details).

The **virtual** command provides the following options:

- **virtual authentication** - Requires users to enter a login name and password to log into the CSS and perform a virtual access (default). The local database is checked in this option.
- **virtual authentication disallowed** - Prevents additional virtual users from logging into the CSS. This selection does not terminate existing connections.

Note To remove users already logged into the CSS, use the **admin-shutdown** command.

- **virtual authentication local-radius** - Checks the local username database for authentication. If local authentication is unsuccessful, the CSS performs a RADIUS server authentication to verify username and password.
- **virtual authentication radius** - Performs a RADIUS server authentication to verify username and password.

- **virtual authentication radius-local** - Performs a RADIUS server authentication to verify username and password. If the RADIUS server authentication is unsuccessful, the CSS checks the local username database for authentication.
- **no virtual authentication** - Does not require users to enter a login name and password to log into the CSS (disables virtual authentication).

The **console** command provides the following options:

- **console authentication** - Requires users to enter a login name and password to log into the CSS console port (default). The local database is checked in this option.
- **console authentication local-radius** - Checks the local username database for authentication. If local authentication is unsuccessful, the CSS performs a RADIUS server authentication to verify username and password.
- **console authentication radius** - Performs a RADIUS server authentication to verify username and password.
- **console authentication radius-local** - Performs a RADIUS server authentication to verify username and password. If the RADIUS server authentication is unsuccessful, the CSS checks the local username database for authentication.
- **no console authentication** - Does not require users to enter a login name and password to log into the CSS console port (disables console authentication).

For example, if an unauthorized user gained access to the CSS:

1. Prevent users from establishing new connections to the CSS by using the **virtual authentication disallowed** command.

   ```
   (config)# virtual authentication disallowed
   ```

2. Terminate all connections using the **admin-shutdown** command.

   ```
   (config)# admin-shutdown
   ```

To display virtual and console authentication settings, use the **show user-database** command (refer to “Showing User Information” in Chapter 1, Logging in and Getting Started).

Restricting Console, FTP, SNMP, Telnet, XML, and Web Management Access to the CSS

Use the **restrict** command to enable or disable console, FTP, SNMP, Telnet, XML, and Web management access to the CSS. Access through a console, FTP, SNMP, and Telnet is enabled by default.

Note Disable Telnet access when you want to use the Secure Shell Host (SSH) server. For information on configuring SSHD, refer to "Configuring Secure Shell Daemon" in Chapter 3, Configuring CSS Network Protocols.

The syntax and options for this global configuration mode command are:

- **restrict console** - Disable console access to the CSS
- **restrict ftp** - Disable FTP access to the CSS
- **restrict snmp** - Disable SNMP access to the CSS
- **restrict telnet** - Disable Telnet access to the CSS
- **restrict XML** - Disable XML access to the CSS
- **restrict web-mgmt** - Disable Web management access to the CSS

To enable access to the CSS:

- **no restrict console** - Enable console access to the CSS
- **no restrict ftp** - Enable FTP access to the CSS
- **no restrict snmp** - Enable SNMP access to the CSS
- **no restrict telnet** - Enable Telnet access to the CSS
- **no restrict xml** - Enable XML access to the CSS
- **no restrict web-mgmt** - Enable Web management access to the CSS

For example, enter:

```
(config)# restrict telnet
```

Finding an IP Address

Use the **find ip address** command to search the CSS configuration for the specified IP address. You can include a netmask for subnet (wildcard) searches. This search can help you avoid IP address conflicts when you configure the CSS.

When you use this command, it checks services, source groups, content rules, ACLs, the management port, syslog, APP sessions, and local interfaces for the specified IP address. If the address is found, the locations of its use are displayed. If no addresses are found, the CSS returns you to the command prompt.

This command is available in all modes. The syntax is:

find ip address *ip_or_host* {*subnet_mask*|**range** *number*}

Enter the:

- IP address in dotted-decimal notation (for example, 192.168.11.1) or enter the host name in mnemonic host-name format (for example, host.domain.com).
- Optional subnet mask as either:
  - A prefix length in CIDR bitcount notation (for example, /24). Do not enter a space to separate the IP address from the prefix length.
  - An IP address in dotted-decimal notation (for example, 255.255.255.0).

  If you enter a mask of 0.0.0.0, the CSS finds all addresses.
- **range** *number* to define how many IP addresses you want to find, starting with the *ip_or_host* address. Enter a number from 1 to 65535. The default range is 1.

  For example, if you enter an IP address of 203.1.1.1 with a range of 10, the CSS tries to find the addresses from 203.1.1.1 through 203.1.1.10.

For example, enter:

```
(config)# find ip address 192.168.0.0

Users of IP address 192.168.0.0
Content Rule - 192.168.12.1, layer 3, owner: lml, state:Active
Content Rule - 192.168.12.1, layer 5, owner: lml, state:Active
Service - 192.168.3.6, serv1, state:Active
Service - 192.168.3.7, serv3, state:Active
Interface - 192.168.1.117. VLAN1
Interface - 192.168.2.117. VLAN1
```

Configuring Flow Parameters

The CSS enables you to configure the following flow parameters using the **flow** command:

- **flow permanent** - Permanent TCP ports that are not reclaimed
- **flow port-reset** - Resets Fast Ethernet and Gigabit Ethernet ports automatically when the CSS detects that they are not responding
- **flow reserve-clean** - Interval flows with port numbers less than or equal to 23 are reclaimed

Configuring Permanent Connections for TCP Ports

The CSS allows you to configure a maximum of ten TCP ports that will have permanent connections and will not be reclaimed by the CSS when the ports are inactive. To configure a TCP port as a permanent connection, use the **flow permanent** command. This command is typically used when load-balancing long-lived connections or you observe the CSS dropping long-lived idle TCP connections.

The options for this command are:

- **flow permanent port1** *portnumber*
- **flow permanent port2** *portnumber*
- **flow permanent port3** *portnumber*
- **flow permanent port4** *portnumber*
- **flow permanent port5** *portnumber*
- **flow permanent port6** *portnumber*
- **flow permanent port7** *portnumber*
- **flow permanent port8** *portnumber*
- **flow permanent port9** *portnumber*
- **flow permanent port10** *portnumber*

Enter a port number from 0 to 65535. The default is 0.

For example, to configure port 1520 as a permanent connection, enter:

```
(config) flow permanent port1 1520
```

To reset a permanent connection to its default port number of 0, use the **no flow permanent** command. For example, to reset the port number for port1 to 0, enter:

```
(config) no flow permanent port1
```

Resetting Fast Ethernet and Gigabit Ethernet Ports

You can program the CSS to reset its associated Fast Ethernet and Gigabit Ethernet ports automatically when it detects that they are not responding during operation. Use the **flow port-reset** command to enable this function. By default, port resetting is enabled on the CSS.

Caution Do not disable port-resets without guidance from Cisco support personnel.

For example, enter:

```
(config)# flow port-reset
```

To disable port resets on the CSS, enter:

```
(config)# no flow port-reset
```

Reclaiming Reserved Telnet and FTP Control Ports

Use the **flow reserve-clean** command in global configuration mode to define how often the CSS scans flows from reserved Telnet and FTP control ports to reclaim them. Control ports have port numbers less than or equal to 23. When the CSS determines that one of these ports has a flow with asymmetrical routing, it reclaims the port.

Enter the **flow reserve-clean** time in seconds as the interval the CSS uses to scan flows. Enter an integer from 0 to 100. The default is 10. To disable the flow reclaiming process, enter a flow reserve-clean value of 0.

For example, enter:

```
(config)# flow reserve-clean 36
```

To disable flow cleanup on Telnet and FTP control ports, enter:

```
(config)# no flow reserve-clean
```

Showing Flow Statistics

Use the **flow statistics** command to display statistics on currently allocated flows.

For example:

```
(config)# flow statistics

Flow Manager Statistics:

                          Current   High     Avg
UDP Flows per second      0         0        0
TCP Flows per second      0         4        0
Total Flows per second    0         4        0
Hits per second           0         0        0

-------------------------------------------------------------
Port      Active      Total         TCP     UDP
-------------------------------------------------------------
1         13          43339169      13      0
2         16          43337519      16      0
5         18          3167362       18      0
6         9           33483528      9       0
```

Configuring Content API

The CSS Content Application Program Interface (API) feature allows you to use a network management workstation to make Web-based configuration changes to the CSS using Extensible Markup Language (XML) documents. XML is a powerful tool that can be used to automatically configure a CSS using all of the CLI commands included in the CSS software, such as to specify server weight and load, to configure load balancing across a group of servers, or to configure content rules to restrict access to a group of directories or files on the servers.

XML code loads a series of CLI commands into the CSS without the need to respond to the prompts, similar to operating in expert mode. As the CSS administrator, plan which type of changes you want to implement and the consequences of these changes as they are performed.

After you create the XML document, you publish (upload) the XML file to the Hypertext Transfer Protocol (HTTP) server embedded in the CSS using a HTTP PUT method.

Creating XML Code

When developing XML code for Content API to issue CLI commands, adhere to the following guidelines. You can use any text editor for creating the XML code.

1. Include the following line as the first line in the XML file:

   ```
   <?xml version="1.0" standalone="yes"?>
   ```

2. Enclose the CLI commands within the <action></action> tag set. For example:

   ```
   <action>add service MyServiceName</action>
   <action>vip address 10.2.3.4</action>
   ```

Note A nested **script play** command (to execute a script line by line from the CLI) is not allowed in an XML file. This restriction is enforced because the actual execution of the XML tag set is performed within a **script play** command

3. Pay attention to mode hierarchy of the CLI commands in the XML file. Each mode has its own set of commands. Many of the modes have commands allowing you to access other related modes. If you enter a series of commands in the improper mode hierarchy, this will result in an XML file that fails to execute properly.

   As an example, the following commands configure an access list (ACL):

```
<?xml version="1.0" standalone="yes" ?>
<config>
    <action>acl 98</action>
        <action>clause 10 permit any any dest any</action>
    <action>apply circuit-(VLAN3)</action>
</config>
```

   In another example, the following commands configure a CSS Ethernet interface:

```
<?xml version="1.0" standalone="yes" ?>
<config>
    <action>interface ethernet-6</action>
    <action>bridge vlan 3</action>
    <action>circuit VLAN3</action>
        <action>ip address 10.10.104.1/16</action>
</config>
```

4. Pay attention to the allowable CLI command conventions for syntax and variable argument in the XML file. If you enter an invalid or incomplete command, this will result in an XML file that fails to execute properly.

Note For overview information on the CLI commands you can use in global configuration mode and its subordinate modes, refer to the *Content Services Switch Command Reference*, Chapter 2, CLI Commands.

XML Document Example

The following example is a complete XML document. The XML document creates three services, an owner, and a content rule, and assigns one of the newly created services to the content rule.

```
<?xml version="1.0" standalone="yes"?>
<config>
    <service name="router">
        <ip_address>10.0.3.1</ip_address>
        <action>active</action>
    </service>
    <service name="sname2">
        <ip_address>10.0.3.2</ip_address>
        <weight>4</weight>
        <action>active</action>
    </service>
    <service name="sname3">
        <ip_address>10.0.3.3</ip_address>
        <weight>5</weight>
        <protocol>udp</protocol>
        <action>suspend</action>
    </service>
    <service name="nick">
        <ip_address>10.0.3.93</ip_address>
        <action>active</action>
    </service>
    <owner name="test">
    <content name="rule">
        <vip_address>10.0.3.100</vip_address>
        <protocol>udp</protocol>
        <port>8080</port>
        <add_service>nick</add_service>
        <action>active</action>
    </content>
    </owner>
</config>
```

Controlling Access to the CSS HTTP Server

To control access to the HTTP server running on the CSS, use the **restrict xml** and **no restrict xml** commands. Clients can send XML documents to this server to configure the CSS. The options for this global configuration mode command are:

- **no restrict xml** - Allow client access to the HTTP server on the CSS.
- **restrict xml** - Deny client access to the HTTP server on the CSS.

Note The **web-mgmt state enable** command (for CSS software version 3.x) performs the same function as the **(config) no restrict xml** command (for CSS software version 4.x) and the **web-mgmt state disable** command performs the same function as the **(config) restrict xml** command. When you use the **web-mgmt state enable** command, it does not appear in the configuration file. Instead, the **(config) no restrict xml** command appears in the configuration file.

Parsing the XML Code

After you complete the XML file, parse the code to ensure that it is syntactically correct. The easiest way to parse XML code is to open the XML file directly from Microsoft® Internet Explorer. Syntax errors are flagged automatically when the file is loaded. If an error occurs, review your XML code and correct all syntax errors.

Publishing the XML Code to the CSS

The completed XML file is remotely published (uploaded) to the HTTP server in the CSS from the external network management workstation by using a HTTP PUT method. The HTTP PUT method uses the IP address of the CSS as the destination URL where you want to publish the XML file.

Note When XML is enabled, the CSS listens for XML connections on port 80.

Note Ensure that the CLI commands in the XML document do not have an impact on the interface configuration through which the XML file transfer process is to occur (for example, including the command **no ip addr 10.1.2.3**, which identifies the IP address of the CSS receiving the XML file). If this occurs, you will disconnect the workstation performing the XML file transfer.

Software is available to simplify the process of publishing XML files to the CSS HTTP server. These software packages offer a simple method to publish files to a Web server. This software uses the HTTP protocol to publish files and require no special software on the Web server side of the connection.

Note An error code in the publishing process usually means that **no restrict xml** (for CSS software version 4.x) or the **webmgmt-state enable** (for CSS software version 3.x) commands have not been issued on the CSS prior to publishing the XML file. See the "Controlling Access to the CSS HTTP Server"section for details.

Testing the Output of the XML Code

Test the output of the XML code by reviewing the running configuration of the CSS. After the XML has been successfully published to the CSS, Telnet to the switch and issue the **show running-config** command to verify that the XML changes have properly occurred. If the XML changes are incorrect or missing, republish the XML code to the CSS as described in the "Publishing the XML Code to the CSS" section.

Configuring the Command Scheduler

Use the **cmd-sched** command to configure the scheduled execution of any CLI commands, including playing scripts. The commands that will be executed are referred to as the command string. To schedule commands, you must create a configuration record, which includes a provision as to when to execute the commands, and the command string.

For example, you can use this command to schedule periodic content replication, the gathering of statistics, and scheduled configuration changes. At the specified time, the command scheduler executes a command string by creating a pseudo-login shell where each string is executed. A cmd-sched record is only scheduled for execution upon completion of its shell. Use the **show lines** command to display information about active pseudo shells (refer to "Showing Current Logins" in Chapter 1, Logging in and Getting Started).

Note To terminate the execution of a command string, use the **disconnect** command.

The syntax and options for this global configuration mode command are:

- **cmd-sched** - Enable command scheduling.
- **cmd-sched record** *name minute hour day month weekday* "*commands...*" {*logfile_name*} - Create a configuration record for the scheduled execution of any CLI commands, including the playing of scripts.

The variables are listed below. When entering minute, hour, day, month, and weekday variables, you may enter a single integer, a wildcard (*), a list separated by commas, or a range separated by a dash (-).

- *name* - The name of the configuration record. Enter an unquoted text string up to 16 characters.
- *minutes* - The minute of the hour to execute this command. Valid numbers are from 0 to 59.
- *hour* - The hour of the day. Valid numbers are from 0 to 23.
- *day* - The day of the month. Valid numbers are from 0 to 31.
- *month* - The month of the year. Valid numbers are from 1 to 12.
- *weekday* - The day of the week. Valid numbers are from 1 to 7. Sunday is 1.

- *command* - The commands you want to execute. Enter a quoted text string up to 255 characters. Separate multiple commands with a semicolon (;) character. If the command string includes quoted characters, use a single quote character; any single quoted characters not preceded by a backslash (\) character is converted to double quotes when the command string is executed.
- *logfile_name*, as an optional variable that defines the name of the log file. Enter a text string up to 32 characters.

Any of the time variables can contain one or some combination of the following values:

- A single number to define a single or exact value for the specified time variable
- A wildcard (*) character matching any valid number for the specified time variable
- A list of numbers separated by commas, up to 40 characters, to define multiple values for a time variable
- Two numbers separated by a dash (-) character indicating a range of values for a time variable

For example, enter:

```
(config)# cmd-sched record periodic_shows 30 21 3 6 1 "show
history;show service;show rule;show system-resources"
```

To enable command scheduler, enter:

```
(config)# cmd-sched
```

To disable command scheduling, enter:

```
(config)# no cmd-sched
```

To delete a configuration record, enter:

```
(config)# no cmd-sched periodic_shows
```

Showing Configured Command Scheduler Records

Use the **show cmd-sched** command to display the state of the command scheduler and information about the records for the scheduled CLI commands. The syntax and options are:

- **show cmd-sched** - Lists the state of the command scheduler and all scheduled CLI command records
- **show cmd-sched name** *record_name* - Lists information about the specified scheduled CLI command record

For example, to view the command scheduler state and all scheduled CLI command records, enter:

```
(config)# show cmd-sched

Cmd Scheduler: Enabled1 record currently configured.

Sched Rec: suspendRule id: 8265b980 Next exec: APR 14 10:46:00
executions:1145
   minList:         0
   hourList:        12
   dayList:         *
   monthList:       *
   weekdayList:     2,3,4,5,6
   cmd:             config;owner owner1;content content1;suspend
```

Table 2-3 describes the fields in the **show cmd-sched** output.

Table 2-3 Field Descriptions for the show cmd-sched Command

Field	Description
Cmd Scheduler	State of the command scheduler (enabled or disabled) and the number of configured records.
Sched Rec	The name of the configuration record.
id	The ID for the record.
next exec	The day and time when the record will be executed.
executions	How many times the record has executed.
minList	The configured minute of the hour to execute the command.
hourList	The configured hour of the day to execute the command.

Table 2-3 Field Descriptions for the show cmd-sched Command (continued)

Field	Description
dayList	The configured day of the month to execute the command.
monthList	The configured month of the year to execute the command.
weekdayList	The configured day of the week to execute the command. Sunday is 1.
cmd	The commands you want to execute. Separate multiple commands with a ; character.

Where to Go Next

Chapter 3, Configuring CSS Network Protocols, describes how to configure the CSS DNS, ARP, RIP, IP, routing, bridging, SSH, and opportunistic Layer 3 forwarding.